



HOW I LEARNED TO
STOP WORRYING
AND
LOVE AI-ASSISTED
CODING TOOLS

Petyo Dimitrov

30th September 2023

About me



- 16 years in Software Engineering
- Senior Software Architect for Musala Soft / Qinshift
- PhD in Computer Science from Technical University Sofia
- Lector of course Distributed Systems in Technical University Sofia

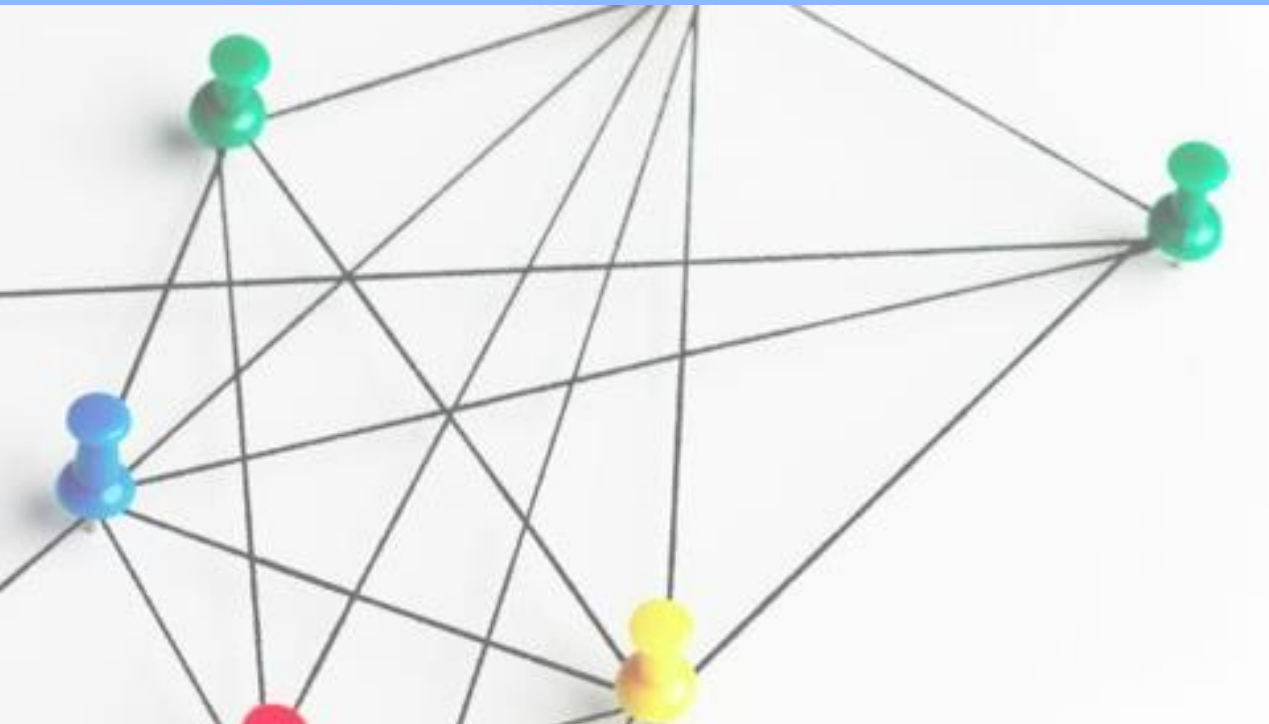
Agenda

- Initial interest
- Reviewed AI-assisted tools
- How I use Copilot(X)?
- How I use ChatGPT?
- How Musala uses AI-assisted tools?
- Recent developments
- Q&A

Prior to 2021



Hi, how can I help you Today!



2022

25
26

24

October
M T W T F S S
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27

2023.03 - GPT 4

2022.11 - ChatGPT

2022.03 - GPT 3.5

2021.10 - GitHub Copilot

2021.08 - Codex

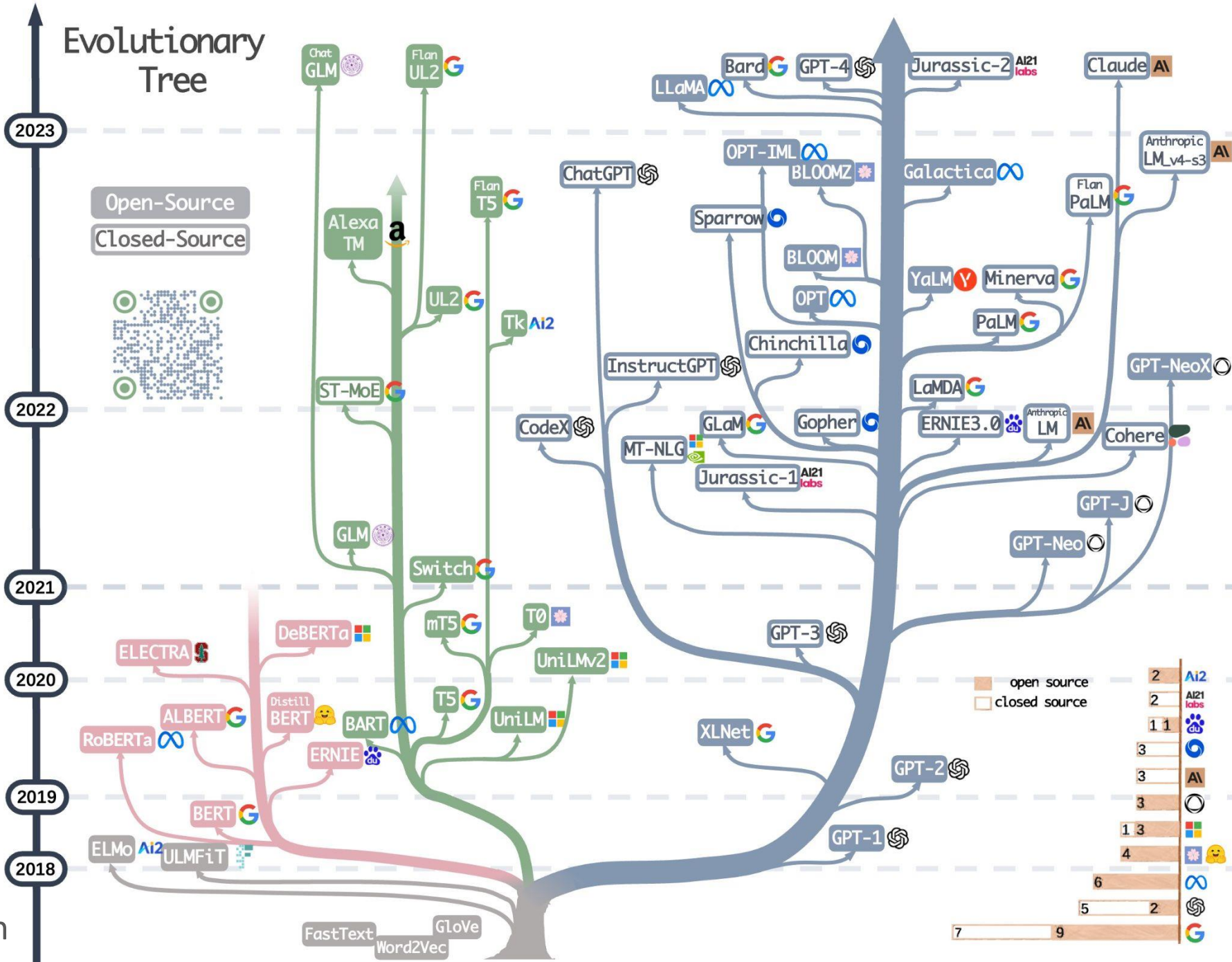
2020.06 - GPT-3 (175B)

2019.02 - GPT-2 (1.5B)

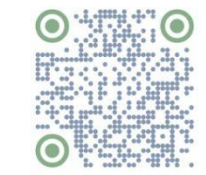
2018.02 - GPT-1 (0.1B)

2017 - Transformer with Attention

Evolutionary Tree



Open-Source
Closed-Source



| | | |
|---|-----------|----|
| 2 | AI21 | |
| 2 | AI21 Labs | |
| 1 | 1 | du |
| 3 | | |
| 3 | | AI |
| 3 | | |
| 1 | 3 | |
| 4 | | |
| 6 | | |
| 5 | 2 | |
| 7 | 9 | |

HackerRank challenge (1)

Asha and Kelly like programming, but Asha practices more than Kelly. Not wanting to fall behind, Kelly resolves to practice more. They each solve a number of problems in a day. Asha has already solved more problems than Kelly. What is the minimum number of days it will take for Kelly to have solved *more* problems than Asha?

For example, if Asha is 5 problems ahead of Kelly and they solve 3 and 5 problems a day, Asha will be ahead by only 3 after the first day, 1 after the second, and Kelly will pass Asha on day 3.

Function Description

Complete the function *minNum* in the editor below. The function must return the minimum number of days needed by Kelly to exceed Asha, or -1 if it's impossible.

minNum has the following parameter(s):

A: Number of problems Asha solves in a day

K: Number of problems Kelly solves in a day

P: Number of problems Asha is ahead to begin

HackerRank challenge (2)

```
public static int minNum(int A, int K, int P) {  
    // If Kelly is not solving more problems per day than Asha, it's impossible  
    if (K <= A) {  
        return -1;  
    }  
  
    int days = 0;  
    while (P > 0) {  
        P = P + A - K; // Calculate the difference after each day  
        days++; // Increment the day counter  
    }  
  
    return days;  
}
```

A = 4, K = 5, P = 1

How we are introducing it in Musala

1. Research & pick most promising candidate tools
2. Define *questionnaire* to assess results
3. Run 1-month trial & collect feedback
4. Research protection mechanisms
5. Identify suitable projects
6. Prepare a list of recommended prompts
7. Require a disclaimer in PR when AI is used
8. Conduct demonstrations and talks to encourage adoption

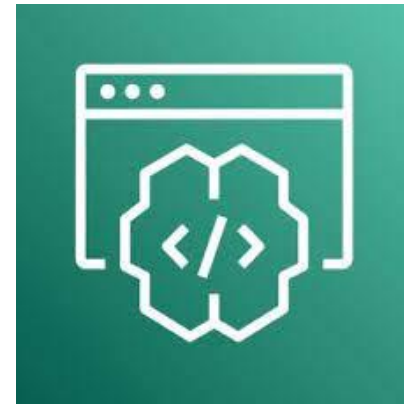
Reviewed tools



GitHub
Copilot



Codeium

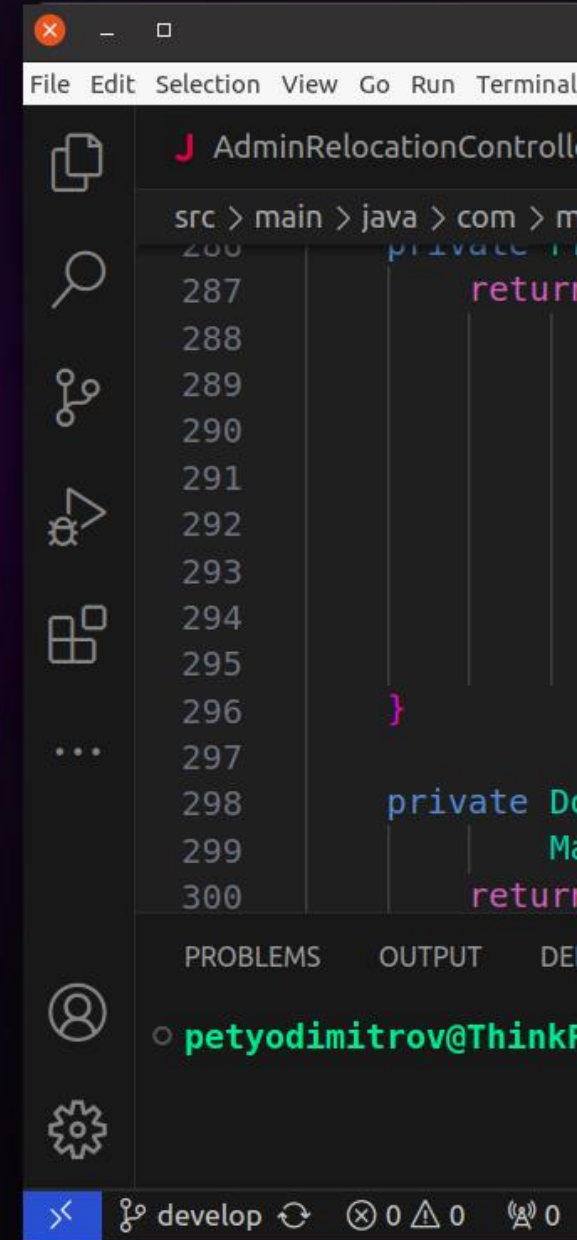


AWS
CodeWhisperer



ChatGPT

How I use Copilot?



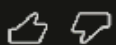
```
EXPLAIN
LANGUAGE TRANSLATION
BRUSHES
TEST GENERATION

la > artemis > modules > practicaltask > controller > AdminPracticalTaskController.java > AdminPracticalTaskController > getAll

48
49 @Operation(summary = "Endpoint to get the practical task")
50 @GetMapping("/{userId}/{type}")
51 public PracticalTaskBaseDTO getPracticalTask(@PathVariable UUID userId, @PathVariable
52     PracticalTaskPrimaryKeyDTO primaryKeyDTO = generatePracticalTaskPrimaryKey(userId)
53
54     PracticalTask practicalTask = practicalTaskService.getByPrimaryKey(primaryKeyDTO)
55     return practicalTaskMapper.practicalTaskToPracticalTaskBaseDTO(practicalTask);
56 }
57
58 @Operation(summary = "Endpoint to get all practical task of user")
59 @GetMapping("/{userId}")
60 public List<PracticalTaskBaseDTO> getAllPracticalTask(@PathVariable UUID userId) {
61     List<PracticalTask> practicalTasks = practicalTaskService.listByPartition(userId)
62
63     // create map with practical task type as key and practical task as value
64     Map<PracticalTaskTypeEnum, PracticalTask> practicalTaskMap = practicalTasks.stream()
65         .collect(Collectors.toMap(PracticalTask::getType, Function.identity()));
66
67     // loop through the list of practical tasks and map them to DTOs
68     return practicalTasks.stream()
69         .map(practicalTaskMapper::practicalTaskToPracticalTaskBaseDTO)
70         .toList();
71 }
72
73 @Operation(summary = "Endpoint to create new practical task")
74 @PostMapping
75 public void createPracticalTask(@RequestBody PracticalTaskCreatedDTO practicalTaskCreatedDTO) {
76     PracticalTask practicalTask = practicalTaskService.create(practicalTaskCreatedDTO)
77     practicalTaskMapper.practicalTaskToPracticalTaskBaseDTO(practicalTask);
78 }
```



GitHub Copilot



This is a Java method that is annotated with `@PostMapping` and maps to the endpoint `/submit/{userId}/{type}`. The method takes in four parameters: `userId`, `type`, `status`, and `submissionType`. `userId` and `type` are path variables, while `status` and `submissionType` are request parameters. The `submitPracticalTask` method generates a `PracticalTaskPrimaryKeyDTO` object using the `userId` and `type` parameters, and then calls the `submitPracticalTaskSync` method of the `practicalTaskService` object, passing in the `PracticalTaskPrimaryKeyDTO` object, `status`, and `submissionType` as arguments.

The purpose of this method is to handle the submission of a practical task by a user. The `userId` parameter is used to identify the user who is submitting the task, while the `type` parameter is used to specify the type of task being submitted. The `status` parameter is used to specify the status of the task submission, with a default value of

```
es > practicaltask > controller > J AdminPracticalTaskController.java > AdminPracticalTaskControll
72         PracticalTask practicalTask = practicalTaskService.createPracticalTask(
73         practicalTaskMapper.practicalTaskToPracticalTaskBaseDTO(practicalTask);
74     }
75
76     @Operation(summary = "Endpoint to submit practical task")
77     @PostMapping("/submit/{userId}/{type}")
78     public void submitPracticalTask(@PathVariable UUID userId,
79                                     @PathVariable PracticalTaskType practicalTaskType,
80                                     @RequestParam(defaultValue = "PENDING") String status,
81                                     @RequestParam(defaultValue = "SUBMISSION") String submissionType,
82                                     PracticalTaskPrimaryKeyDTO primaryKeyDTO = generatePracticalTaskPrimaryKeyDTO(userId, practicalTaskType),
83                                     practicalTaskService.submitPracticalTaskSync(primaryKeyDTO, status, submissionType);
84     }
85
86     @Operation(summary = "Endpoint to delete a practical task")
87     @DeleteMapping("/{userId}/{type}")
88     public PracticalTaskBaseDTO deletePracticalTask(@PathVariable UUID userId,
89                                                     @PathVariable PracticalTaskType practicalTaskType,
90                                                     PracticalTaskPrimaryKeyDTO primaryKeyDTO = generatePracticalTaskPrimaryKeyDTO(userId, practicalTaskType),
91                                                     PracticalTask practicalTask = practicalTaskService.deletePracticalTask(primaryKeyDTO, practicalTaskType);
92     return practicalTaskMapper.practicalTaskToPracticalTaskBaseDTO(practicalTask);
93     }
94
95     @Operation(summary = "Endpoint to start practical task")
96     @PostMapping("/start/{userId}/{type}")
97     public PracticalTaskBaseDTO startPracticalTask(@PathVariable UUID userId,
98                                                     @PathVariable PracticalTaskType practicalTaskType,
99                                                     PracticalTask practicalTask = practicalTaskService.startPracticalTask(primaryKeyDTO, practicalTaskType);
100     return practicalTaskMapper.practicalTaskToPracticalTaskBaseDTO(practicalTask);
101     }
```

> main > java > com > musala > artemis > modules > practicaltask > controller > AdminPracticalTaskController.java > AdminPracticalTaskController.java

```

41 @RequiredArgsConstructor
42 @RequestMapping("/practical-tasks")
43 @Tag(name = "Practical Task Controller", description = "Endpoints handling admin operation")
44 @FeatureToggle(FeatureToggleEnum.BE_PRACTICAL_TASK)
45 public class AdminPracticalTaskController {

```

```

46
47     private final PracticalTaskService practicalTaskService;
48     private final PracticalTaskMapper practicalTaskMapper = Mappers.getMapper(clazz:PracticalTaskMapper.class);

```

```

49
50     @Operation(summary = "Endpoint to get the practical task")
51     @GetMapping("/{userId}/{type}")
52     public PracticalTaskBaseDTO getPracticalTask(@PathVariable UUID userId, @PathVariable PracticalTaskTypeEnum type) {
53         PracticalTaskPrimaryKeyDTO primaryKeyDTO = generatePracticalTaskPrimarykey(userId, type);
54
55         PracticalTask practicalTask = practicalTaskService.getByPrimaryKey(primaryKeyDTO);
56         return practicalTaskMapper.practicalTaskToPracticalTaskBaseDTO(practicalTask);
57     }

```

```


58
59     @Operation(summary = "Endpoint to get all practical task of user")
60     @GetMapping("/{userId}")
61     public List<PracticalTaskBaseDTO> getAllPracticalTask(@PathVariable UUID userId) {
62         List<PracticalTask> practicalTasks = practicalTaskService.listByPartition(userId);
63
64         return practicalTasks.stream()
65             .map(practicalTaskMapper::practicalTaskToPracticalTaskBaseDTO)
66             .toList();
67     }

```


```

68
69     @Operation(summary = "Endpoint to create new practical task")
70     @PostMapping

```

 petyodimitrov-musala

generate unit test

 GitHub Copilot



Here's an example of a unit test for the
getPracticalTask method:

```

@Test
void getPracticalTask_ReturnsPracticalTaskBaseDTO() {
    UUID userId = UUID.randomUUID();
    PracticalTaskTypeEnum type = PracticalTaskTypeEnum.PRIMARY_KEY;
    PracticalTaskPrimarykeyDTO primaryKeyDTO = generatePracticalTaskPrimarykey(userId, type);
    PracticalTask practicalTask = practicalTaskService.getByPrimaryKey(primaryKeyDTO);
    PracticalTaskBaseDTO expectedPracticalTaskBaseDTO = practicalTaskMapper.practicalTaskToPracticalTaskBaseDTO(practicalTask);

    when(practicalTaskService.getByPrimaryKey(primaryKeyDTO)).thenReturn(practicalTask);

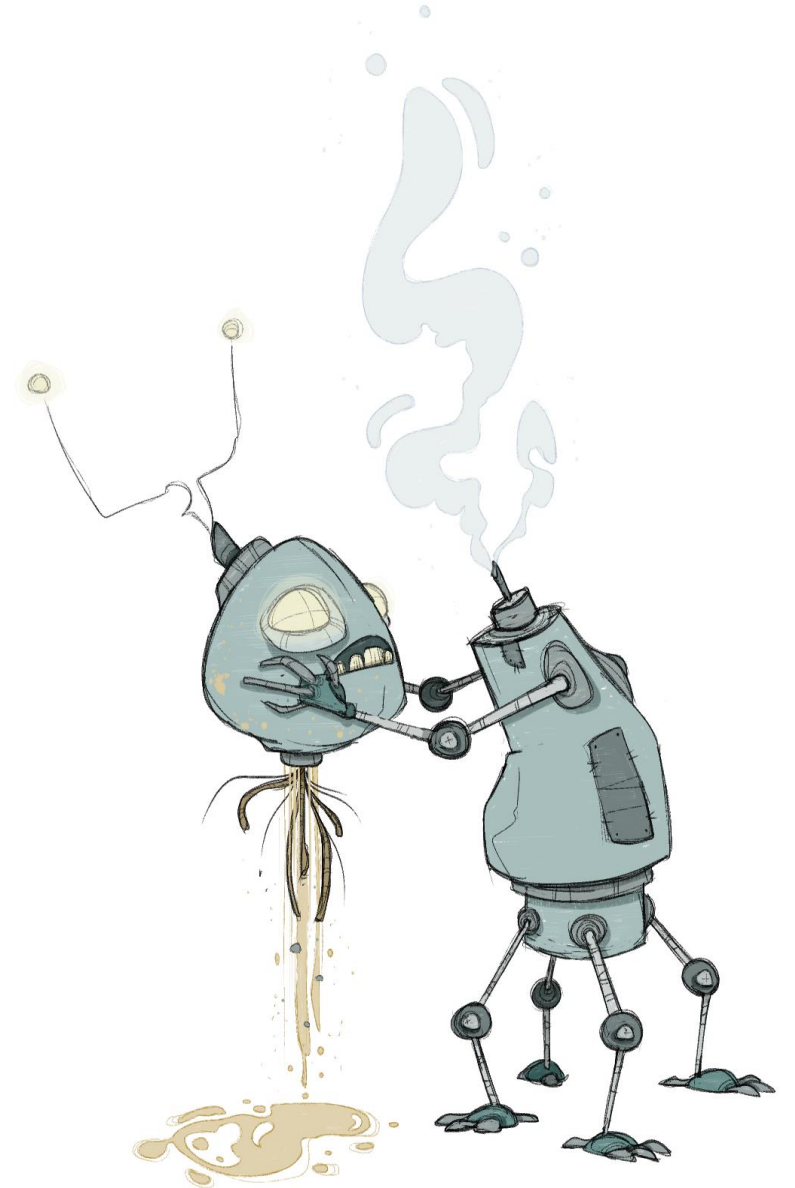
    mockMvc.perform(get(ADMIN_PRACTICAL_TASKS_URL + "/{userId}/{type}", userId, type)
        .contentType(MediaType.APPLICATION_JSON)
        .andExpect(status().isOk())
        .andExpect(content().contentType(MediaType.APPLICATION_JSON))
        .andExpect(jsonPath("$.id").value(practicalTask.getId()))
        .andExpect(jsonPath("$.userId").value(practicalTask.getUserId()))
        .andExpect(jsonPath("$.type").value(practicalTask.getType().name()))
        .andExpect(jsonPath("$.primaryKey").value(practicalTask.getPrimaryKey().name()))
        .andExpect(jsonPath("$.description").value(practicalTask.getDescription()))
        .andExpect(jsonPath("$.createdAt").value(practicalTask.getCreatedAt().toString()))
        .andExpect(jsonPath("$.updatedAt").value(practicalTask.getUpdatedAt().toString()))
        .andExpect(jsonPath("$.deleted").value(practicalTask.isDeleted()))
        .andExpect(jsonPath("$.deletedAt").value(practicalTask.getDeletedAt().toString()))
        .andExpect(jsonPath("$.deletedBy").value(practicalTask.getDeletedBy().name()))
        .andExpect(jsonPath("$.deletedAt").value(practicalTask.getDeletedAt().toString()))
        .andExpect(jsonPath("$.deletedBy").value(practicalTask.getDeletedBy().name()))
    );
}

```

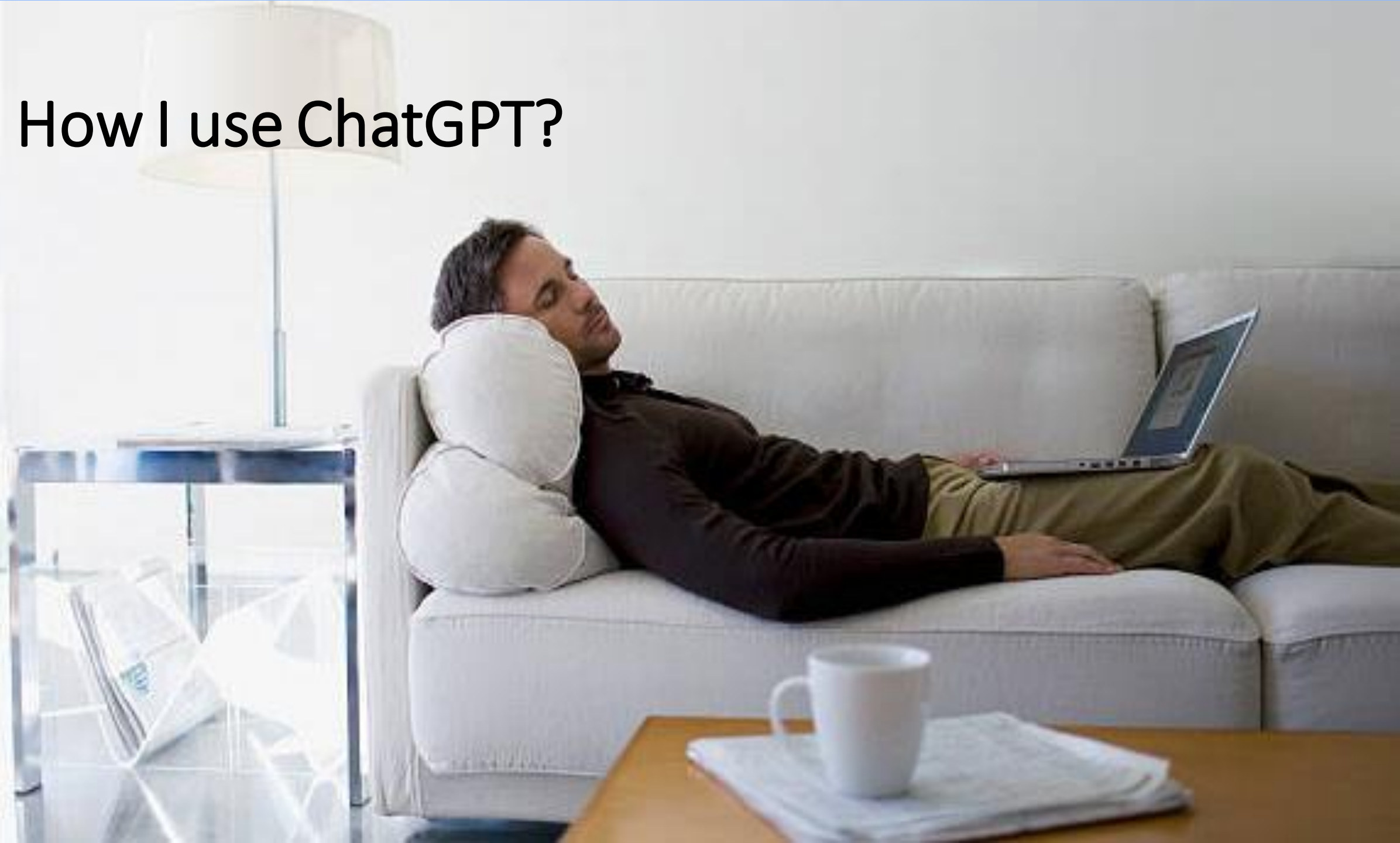
What is the purpose of unit testing?

Weaker features of Copilot

- Inline fixing
- Code translation
- Refactoring



How I use ChatGPT?



For inspiration



As a Rubber ducky



Writing code in
"second" languages

```
ws.on("message", m => {  
  let a = m.split(" ")  
  switch(a[0]){  
    case "connect":  
      if(a[1]){  
        if(clients.has(a[1])){  
          ws.send("connected");  
          ws.id = a[1];  
        }else{  
          ws.id = a[1]  
          clients.set(a[1], {client: position(a[1], clients)})  
          ws.send("connected")  
        }  
      }else{  
        ws.send("connected")  
      }  
    }  
  }  
})
```

Analyze REST APIs (1)

Practical Task Controller Endpoints handling admin operations for User Practical task



| | | | | |
|--------|--|--|---|---|
| POST | <code>/api/admin/practical-tasks</code> | Endpoint to create new practical task | ✓ | 🔒 |
| GET | <code>/api/admin/practical-tasks/{userId}/{type}</code> | Endpoint to get the practical task | ✓ | 🔒 |
| POST | <code>/api/admin/practical-tasks/{userId}/{type}</code> | Endpoint to resend practical task submission message to the Hunter | ✓ | 🔒 |
| DELETE | <code>/api/admin/practical-tasks/{userId}/{type}</code> | Endpoint to delete a practical task | ✓ | 🔒 |
| POST | <code>/api/admin/practical-tasks/submit/{userId}/{type}</code> | Endpoint to submit practical task | ✓ | 🔒 |
| POST | <code>/api/admin/practical-tasks/start/{usrId}/{type}</code> | Endpoint to start practical task | ✓ | 🔒 |
| GET | <code>/api/admin/practical-tasks/{userId}</code> | Endpoint to get all practical task of user | ✓ | 🔒 |

Analyze REST APIs (2)

Problems with the API

1. RESTful Violations: The current API design is not entirely RESTful. A RESTful approach would separate different functionalities into distinct endpoints, based on HTTP methods. However, here, both "get the practical task" and "resend practical task submission" operations use the same path but with different HTTP methods (GET and POST respectively), which can lead to confusion.
2. Ambiguity in Path Parameters: There's inconsistency in the 'userId' parameter in the last endpoint. It's 'usrId' instead of 'userId', which could be confusing.

Analyze REST APIs (3)

CRUD Completeness

This API is CRUD complete. It allows for the creation (POST), reading (GET), updating (potentially via POST), and deletion (DELETE) of resources.

Use Cases Fulfillment

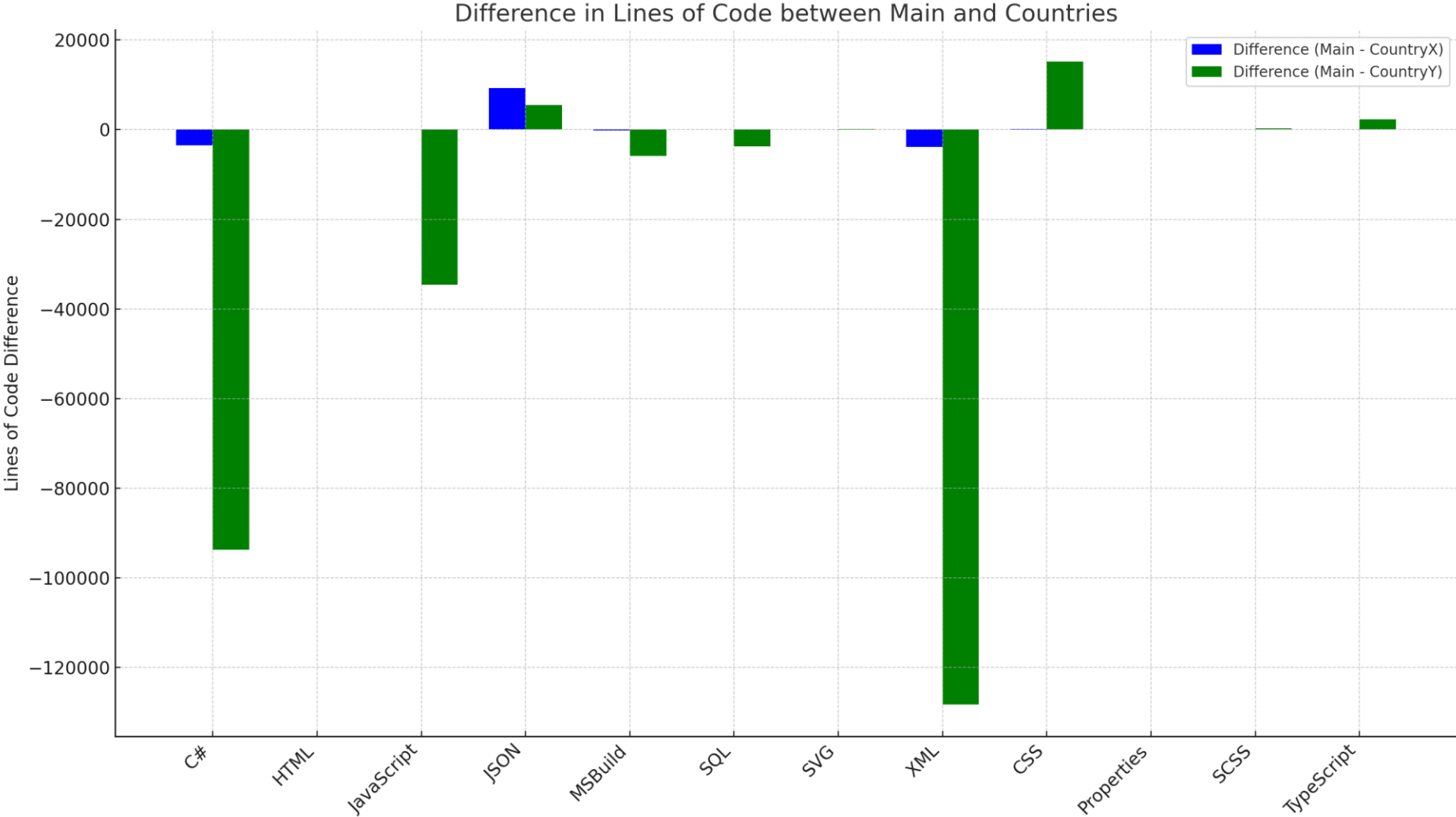
1. "I want to find the practical task of a user with a specific type and start it"
 - Yes. Use `GET /api/admin/practical-tasks/{userId}/{type}` to find the practical task, and `POST /api/admin/practical-tasks/start/{userId}/{type}` to start it.
2. "I want to pause a running practical task"
 - No. There is no endpoint to pause a running practical task.

Advanced Data Analysis (1)

| Code | Main | CountryX | CountryY |
|------------|---------|----------|----------|
| C# | 134,064 | 137,548 | 227,826 |
| HTML | 1,126 | 1,126 | 1,126 |
| JavaScript | 191,082 | 191,082 | 225,657 |
| JSON | 16,514 | 7,220 | 11,072 |
| MSBuild | 14,919 | 15,063 | 20,812 |

Display the difference between Main and CountryX and CountryY for the different code types. The numerical values are the number of lines of code.

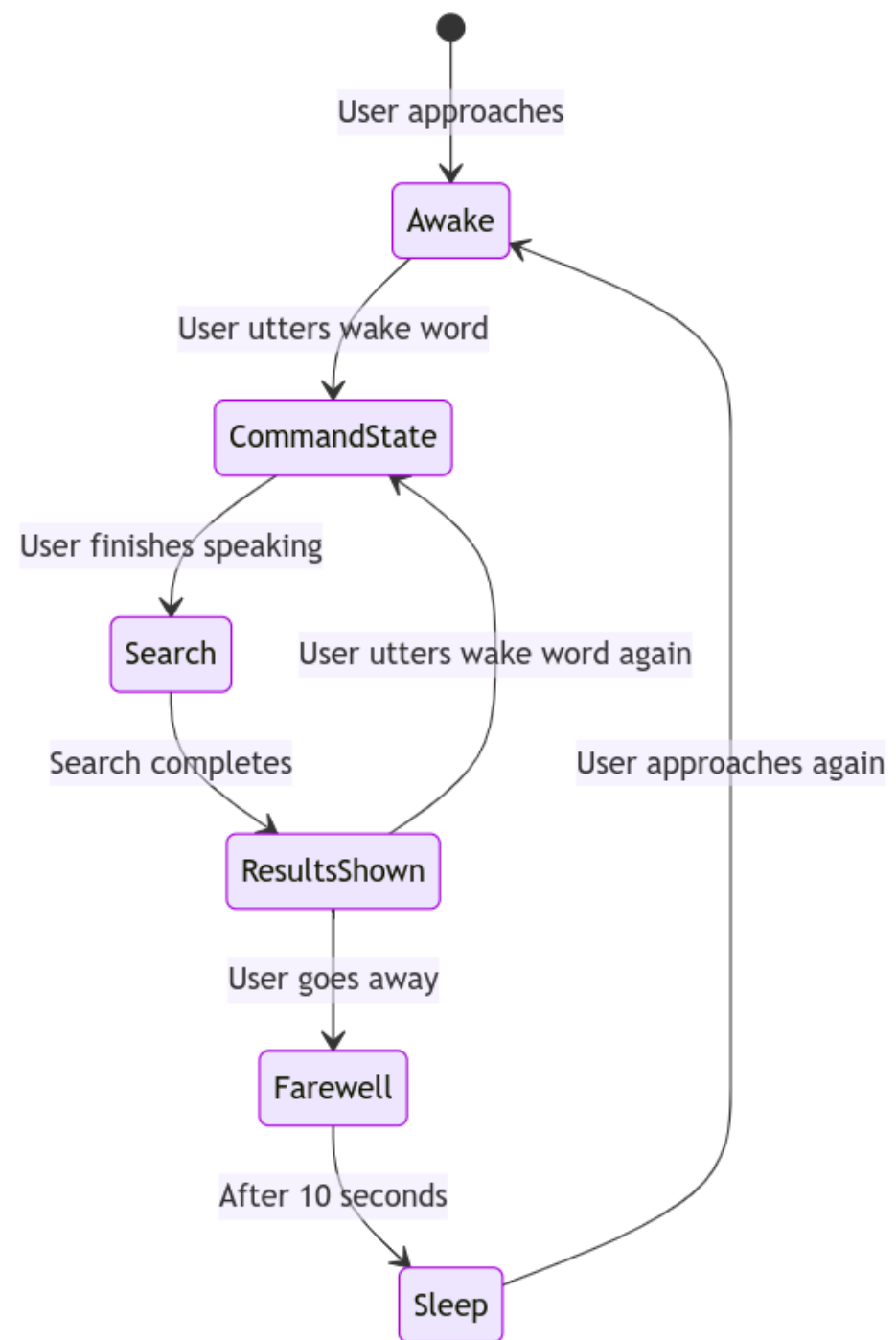
Advanced Data Analysis (2)



Create diagrams

stateDiagram-v2

```
1  [*] --> Awake: User approaches
2  Awake --> CommandState: User utters wake word
3
4  CommandState --> Search: User finishes speaking
5
6  Search --> ResultsShown: Search completes
7
8  ResultsShown --> CommandState: User utters wake word again
9  ResultsShown --> Farewell: User goes away
10
11
12  Farewell --> Sleep: After 10 seconds
13
14  Sleep --> Awake: User approaches again
```



Generating presentations

Generate Python code for creating a PowerPoint presentation on the topic "*Basic Python syntax and data types*". Fill in the details with your knowledge. I need 7 slides. Run the code and return the resulting presentation.

Generating presentations

Composite Data Types

- - List (list)
- - Tuple (tuple)
- - Dictionary (dict)
- - Set (set)

Prompt engineering - Examples or reference



Prompt engineering - Adopt a persona & style

A person wearing a dark suit, white shirt, and red tie is holding a white, featureless mask in front of their face. The person's hands are visible, holding the mask from the sides. The background is dark and out of focus.

Custom Instructions / Roles

Prompt engineering - Step by step



Prompt engineering - Ask questions



Prompt engineering - Reflection



Prompt engineering - Delimiters



Prompt engineering - Periodic summary*



Effects on productivity

~ +5-10%

| Productivity metric | % improvement (before/after GitHub CoPilot use) | Weighting | Productivity Impact Assessment (weighted) |
|--------------------------|---|-----------|---|
| Throughput | +10% | 30% | +3% |
| Cycle Time | +5% | 30% | +1.5% |
| Escaped Defects | +0% | 20% | +0% |
| Sprint Target Completion | +4% | 10% | +0.4% |
| eNPS | +4% | 10% | +0.4% |
| Weighted Average | | | +5.3% |

Musala AI-related projects

- Commercial: **Retrieval-Augmented Generation (RAG)**
- Internal: **Summarization** (interview)
- Internal: **Code review** (practical tasks)
- Approach:
 - ChatBot
 - API
 - Local model*

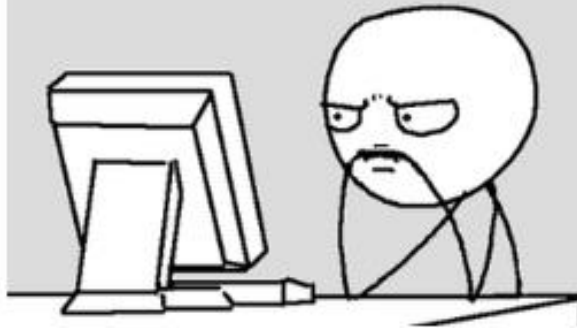


Risks - Devs

- Verbose interactions
- Hallucinations
- Overreliance on the machine

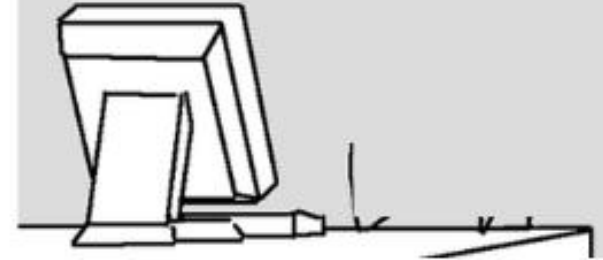
Before Chat GPT

* Developer coding - 2 hours

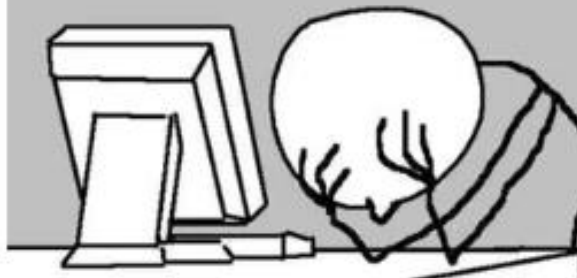


After Chat GPT

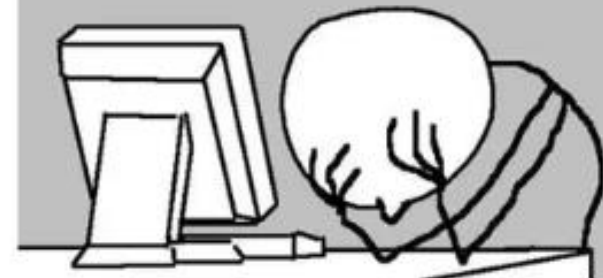
* ChatGPT generating code - 5 min



* Developer debugging - 6 hours



* Developer debugging - 24 hours



Risks - Data

- Leaking proprietary data
- Intellectual Property
- Siloed solutions by large vendors



Risks - EU AI Act



Social credit,
realtime biometric



Critical, education,
law, employment



Chatbots, image
processing



Spam filters,
computer games

Latest news

November – AWS Gen AI

November – 365 Copilot AI Assistant

September – Minstral

September – ChatGPT Voice & Images

September – Copilot Chat*

August – Enterprise ChatGPT

August – Code Llama (Instruct, Python)

July – Llama2

July – Claude2

July – ChatGPT Code Interpreter



Conclusion

- A new tool at our disposal like Google & StackOverflow
- Only as an assistant
- Juniors, Mediors and Seniors benefit differently
- Copilot's advantage is seamless IDE integration
- ChatGPT helps with a wider variety of tasks
- There are risks & challenges ahead



Questions



Thanks!



[www.linkedin.com/in/
petyo-dimitrov](https://www.linkedin.com/in/petyo-dimitrov)



petyo.dimitrov@musala.com
petyo.dimitrov@gmail.com